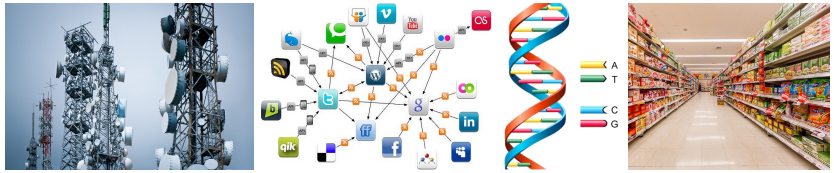
An Efficient Algorithm for SPM with CP

J. AOGA[1], T. Guns[2], P. Schaus[1]

[1]UCLouvain, [2]KULeuven — Belgium

ECML PKDD 2016, Riva del Garda, Italy, 19–23/09/2016

Problem of Sequential Pattern Mining (SPM)



Client1	Milk	**Coffee**	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

- Sequence Database (SDB)

Sequence

Client1	Milk	**Coffee**	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

- Sequence Database (SDB)
- Sequence : < Milk Coffee Sugar Coffee Sugar>

Sub sequence — Sequence

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

- Sequence Database (SDB)
- Sequence : < Milk Coffee Sugar Coffee Sugar>
- Subsequence : <Coffee Sugar>

Sub sequence — Sequence

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

- Sequence Database (SDB)
- Sequence : < Milk Coffee Sugar Coffee Sugar>
- Subsequence : <Coffee Sugar>
- Support (<Coffee Sugar>) = 3

Sub sequence — Sequence

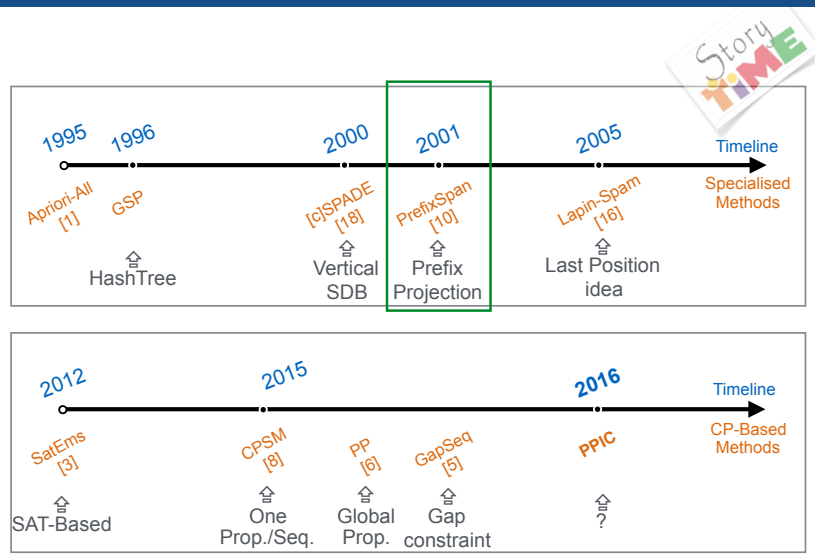
Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

- Sequence Database (SDB)
- Sequence : < Milk Coffee Sugar Coffee Sugar>
- Subsequence : <Coffee Sugar>
- Support (<Coffee Sugar>) = 3

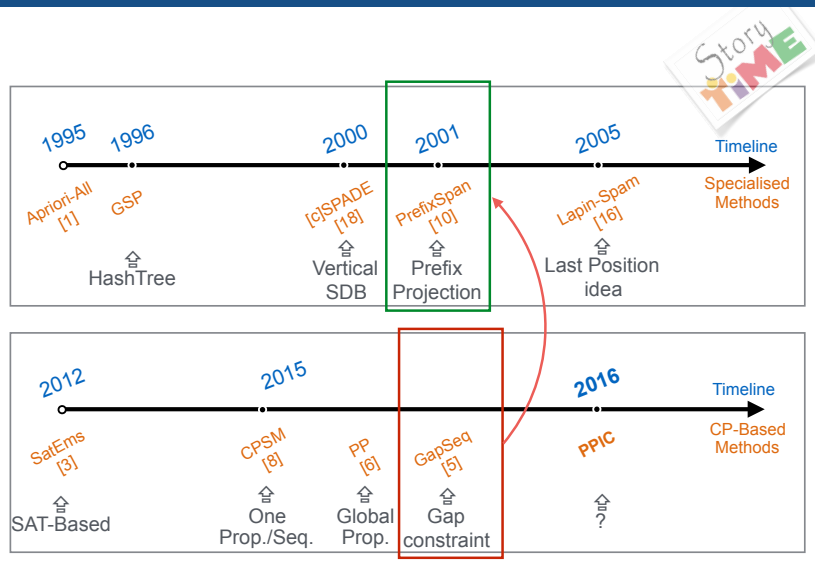
Problem : Find all subsequences with support ≥ Given Threshold

Related Work

Related Work



Related Work

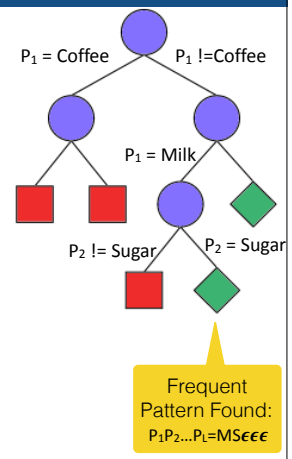


CP : Filtering + DFSearch

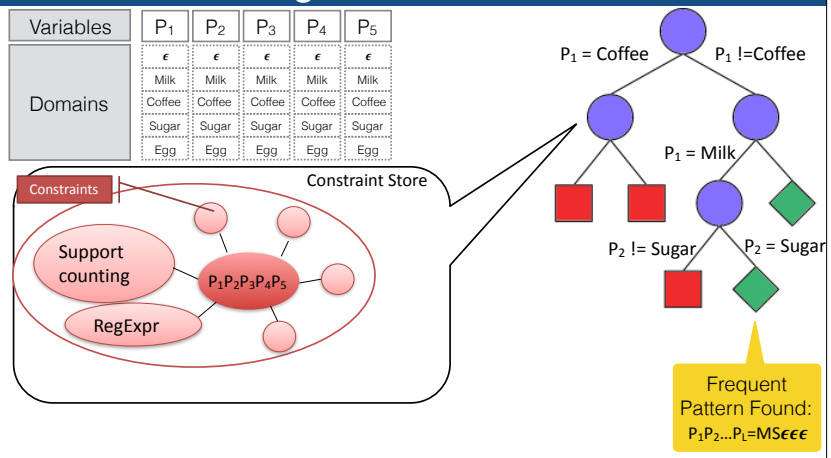
Variables	P_1	P_2	P_3	P_4	P_5
Domains	ϵ	ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg

CP : Filtering + DFSearch

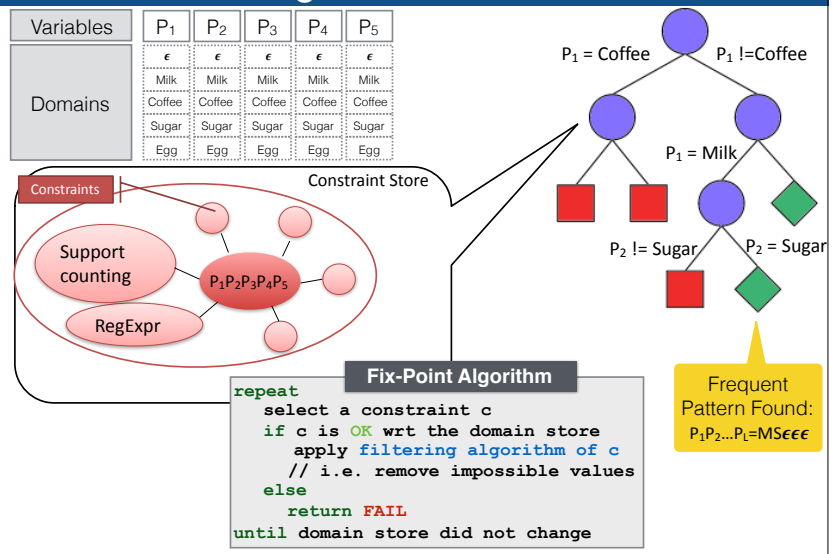
Variables	P_1	P_2	P_3	P_4	P_5
Domains	ϵ	ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



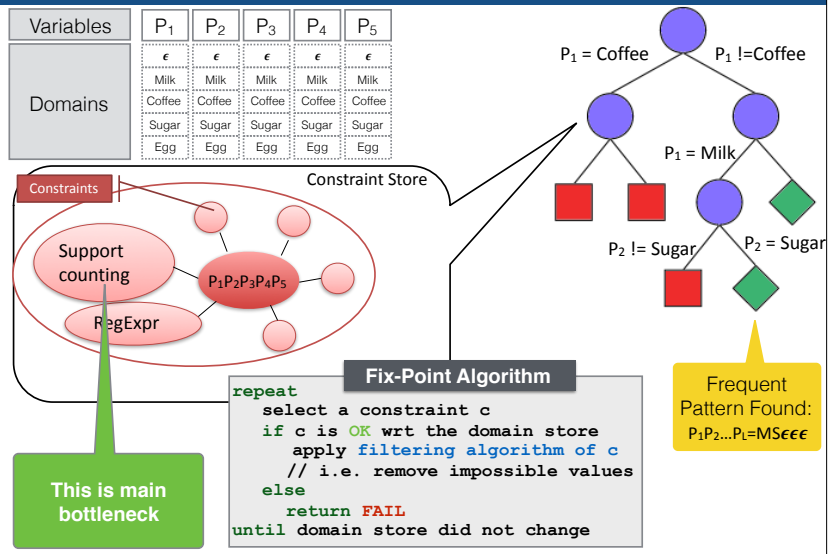
CP : Filtering + DFSearch



CP : Filtering + DFSearch



CP : Filtering + DFSearch



Improvements of Literature (1/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ?** => **reversible vectors** making use of **trailing techniques.**

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports M :

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
E : 1

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
E : 1

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

0 1 2 3 4 5 6 7 8 9 10 11 12 13

7

Given
Threshold=3
(75%)

M

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

0 1 2 3 4 5 6 7 8 9 10 11 12 13

7

Given
Threshold=3
(75%)

M

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

0 1 2 3 4 5 6 7 8 9 10 11 12 13

7

Given
Threshold=3
(75%)

M

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

0 1 2 3 4 5 6 7 8 9 10 11 12 13

7

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
		Sugar	Sugar	Sugar	Sugar
		Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

7

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
		Sugar	Sugar	Sugar	Sugar
		Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

7

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
		Sugar	Sugar	Sugar	Sugar
		Egg	Egg	Egg	Egg

start=4
Size=3

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

7

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
		Sugar	Sugar	Sugar	Sugar
		Egg	Egg	Egg	Egg

Top of the sub-stack

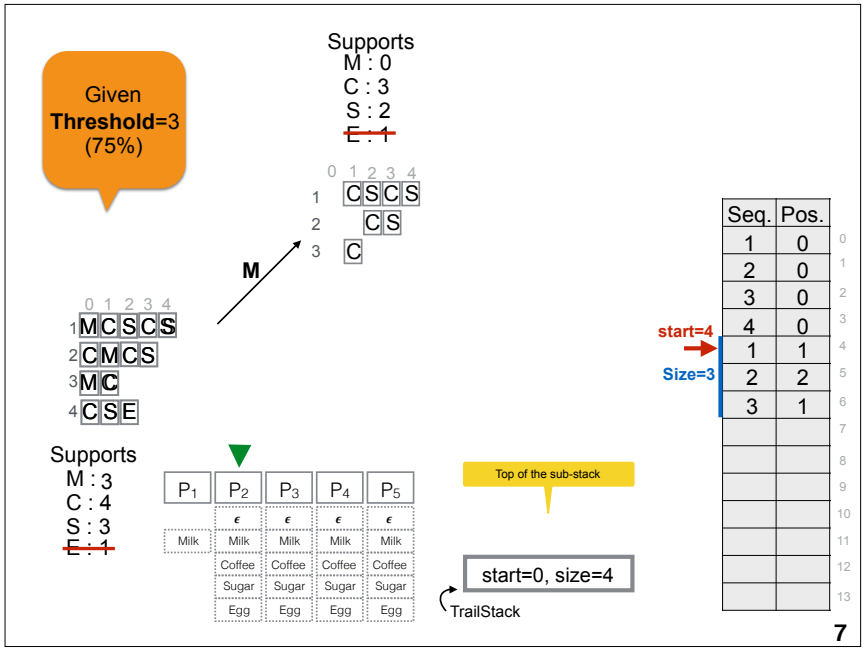
start=0, size=4

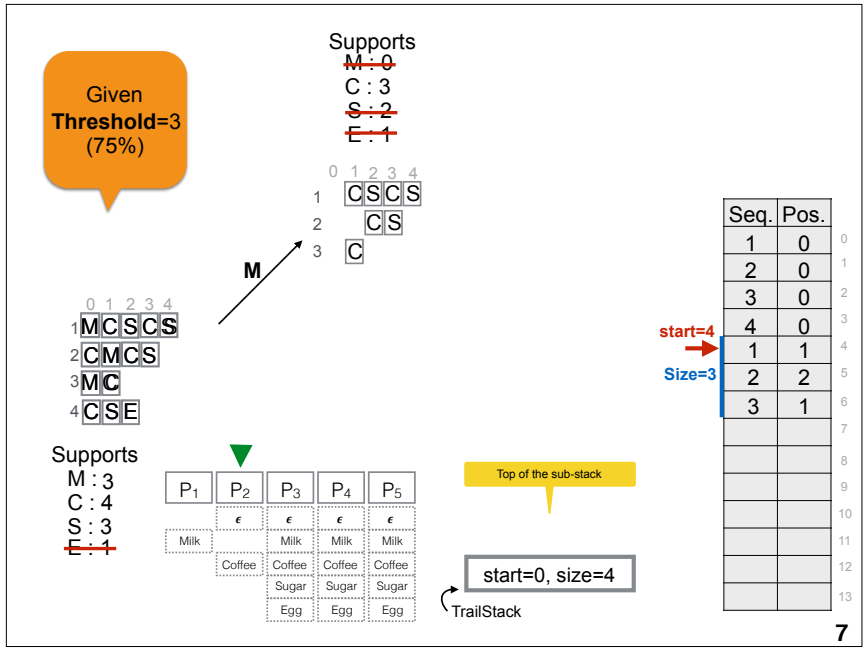
TrailStack

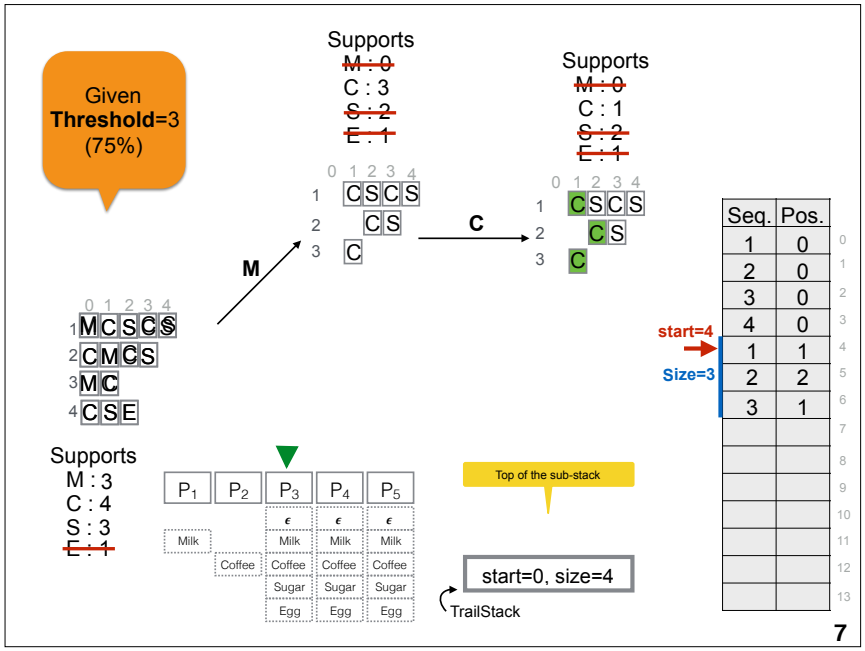
start=4
Size=3

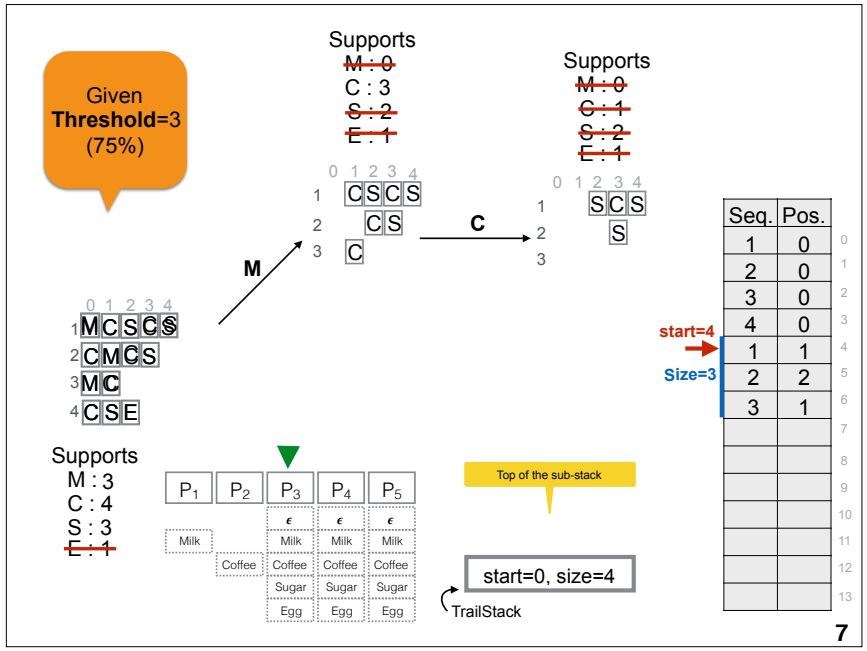
Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

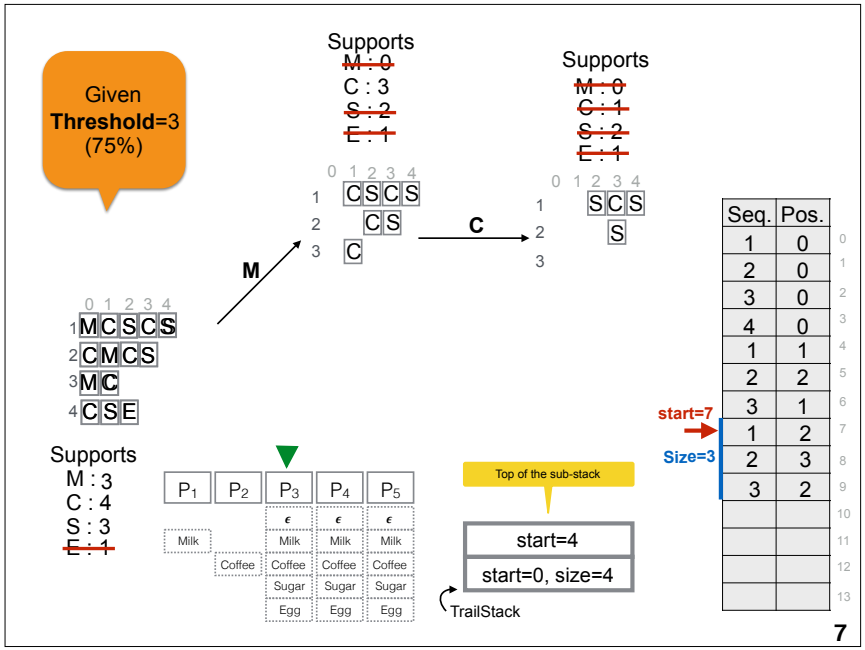
7

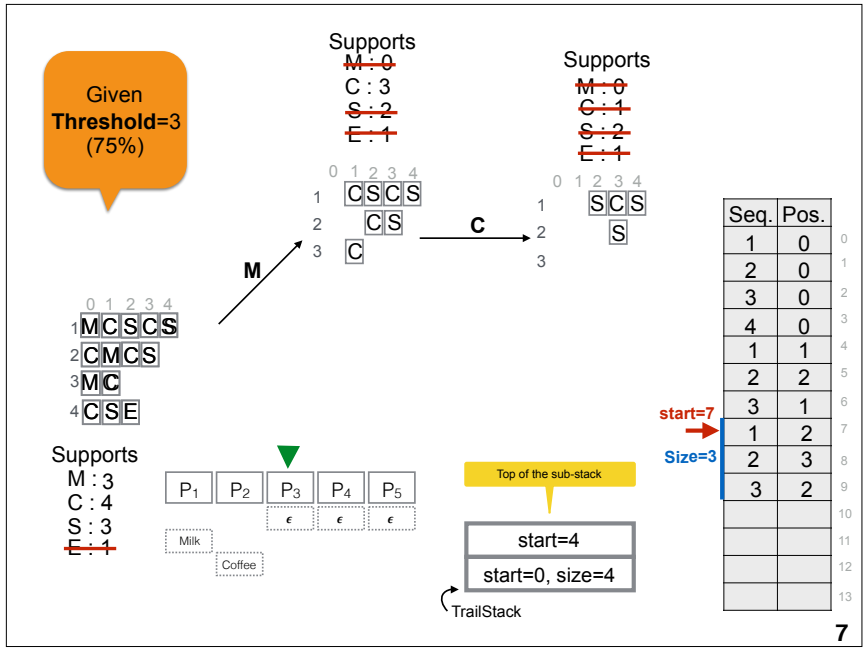

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
~~E : 1~~
M
Supports
M : 3
C : 4
S : 3
~~E : 1~~
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
7

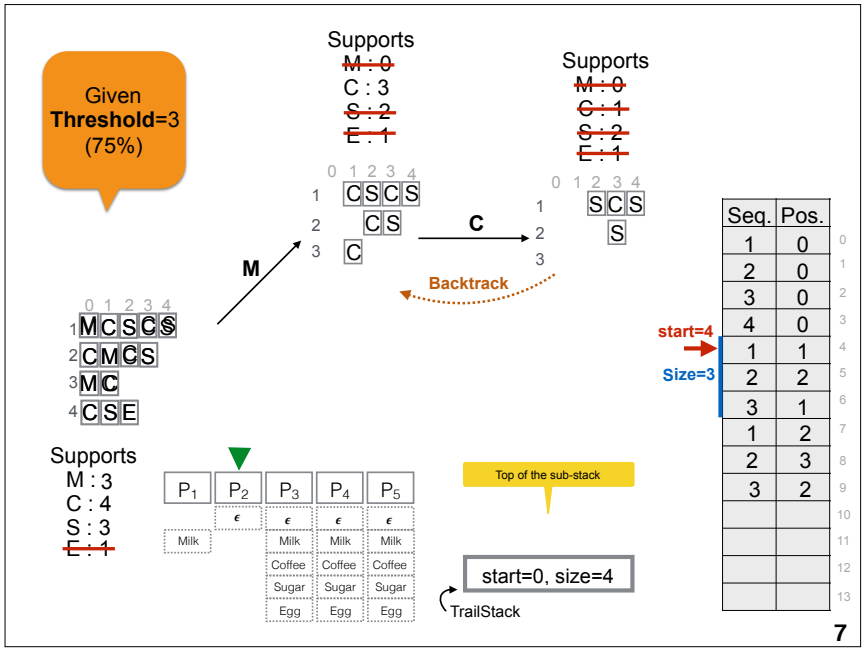

Given
Threshold=3
(75%)
Supports
~~M : 0~~
C : 3
~~S : 2~~
~~E : 1~~
M
Supports
M : 3
C : 4
S : 3
~~E : 1~~
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
7

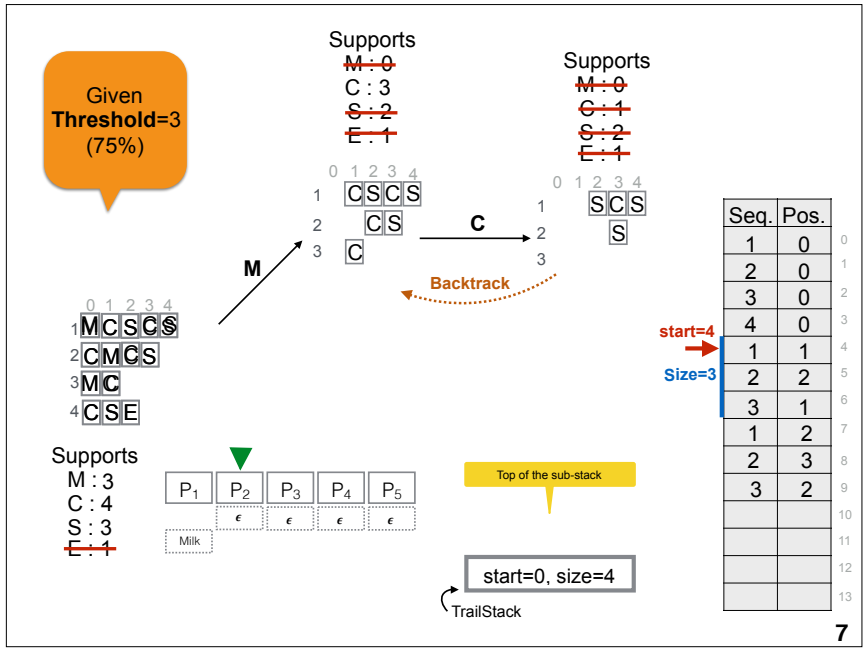

Given
Threshold=3
(75%)
Supports
~~M : 0~~
C : 3
~~S : 2~~
~~E : 1~~
M
C
Supports
~~M : 0~~
C : 1
~~S : 2~~
~~E : 1~~
Supports
M : 3
C : 4
S : 3
~~E : 1~~
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
7

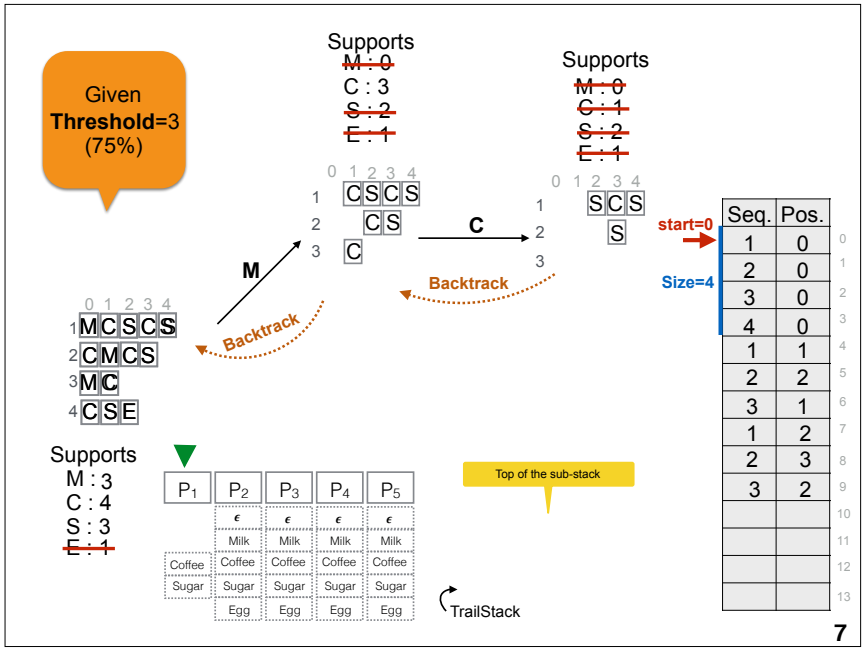

Given
Threshold=3
(75%)
Supports
~~M : 0~~
C : 3
~~S : 2~~
~~E : 1~~
M
C
Supports
~~M : 0~~
~~C : 1~~
~~S : 2~~
~~E : 1~~
Supports
M : 3
C : 4
S : 3
~~E : 1~~
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
7


Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Coffee
Sugar
Egg
Top of the sub-stack
start=4
start=0, size=4
TrailStack
start=7
Size=3
Seq. Pos.
7

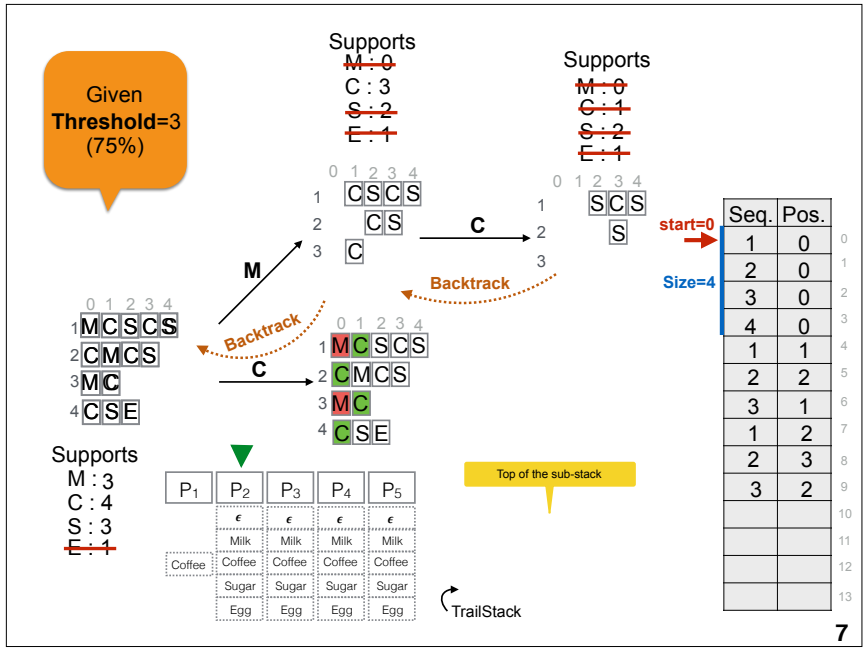

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Coffee
Top of the sub-stack
start=4
start=0, size=4
TrailStack
start=7
Size=3
Seq. Pos.
7


Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Coffee
Sugar
Egg
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
Seq. Pos.
7

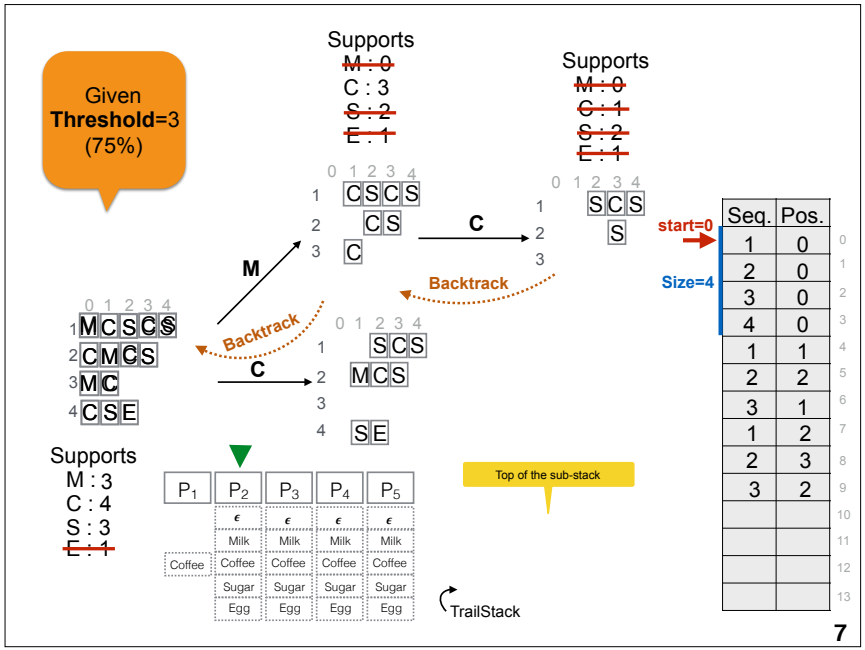

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
Seq. Pos.
7



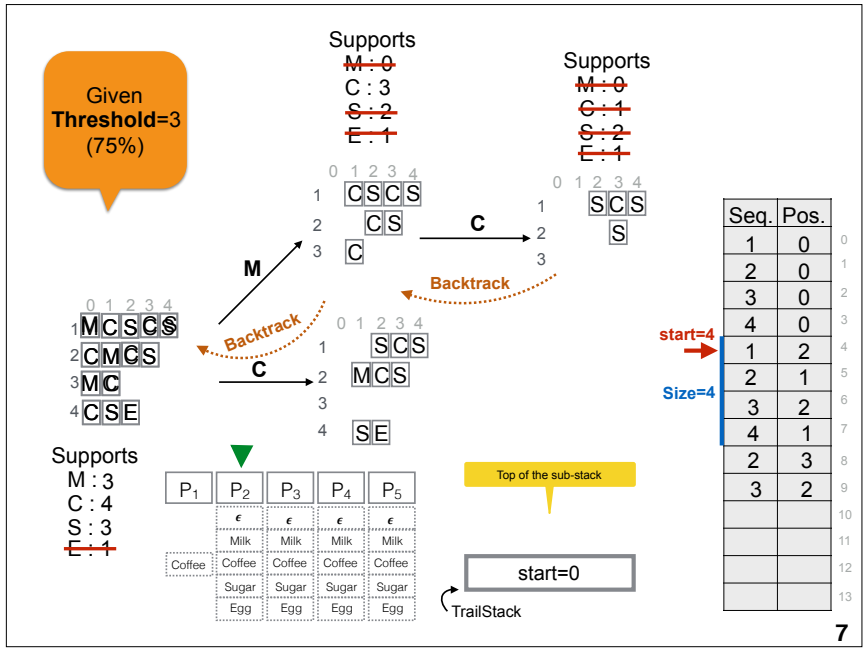
	Seq.	Pos.
0	1	0
1	2	0
2	3	0
3	4	0
4	1	1
5	2	2
6	3	1
7	1	2
8	2	3
9	3	2
10		
11		
12		
13		



	Seq.	Pos.
0	1	0
1	2	0
2	3	0
3	4	0
4	1	1
5	2	2
6	3	1
7	1	2
8	2	3
9	3	2
10		
11		
12		
13		



	Seq.	Pos.
0	1	0
1	2	0
2	3	0
3	4	0
4	1	1
5	2	2
6	3	1
7	1	2
8	2	3
9	3	2
10		
11		
12		
13		



	Seq.	Pos.
0	1	0
1	2	0
2	3	0
3	4	0
4	1	2
5	2	1
6	3	2
7	4	1
8	2	3
9	3	2
10		
11		
12		
13		

Improvements of Literature (2/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ? => reversible vectors** making use of **trailing techniques.**

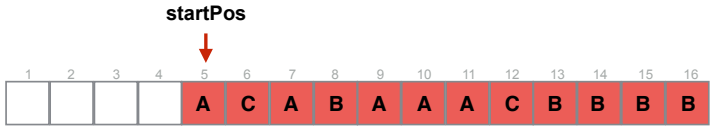
2. **Support Count Improvement. How to compute support efficiently?** Visit **only the last position** of each symbol after **start position.**

How to compute items support in SDB?

★ Identify items which exist in each sequence of SDB and increase items support value.

► **Often many repeating symbols:** cache for each symbol its 'last position' and only iterate over those (**O(m) vs O(n)**)

✦ **startPos**=5

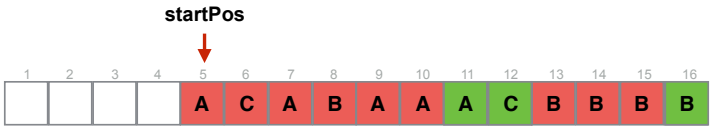


How to compute items support in SDB?

★ Identify items which exist in each sequence of SDB and increase items support value.

► **Often many repeating symbols:** cache for each symbol its 'last position' and only iterate over those (**O(m) vs O(n)**)

✦ **startPos**=5



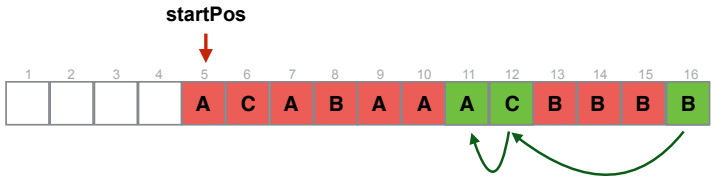
★ **Last Position List** = [(B,16),(C,12),(A,11),(D,2),(E,0)]

How to compute items support in SDB?

★ Identify items which exist in each sequence of SDB and increase items support value.

► **Often many repeating symbols:** cache for each symbol its 'last position' and only iterate over those (**O(m) vs O(n)**)

✦ **startPos**=5



★ **Last Position List** = [(B,16),(C,12),(A,11),(D,2),(E,0)]

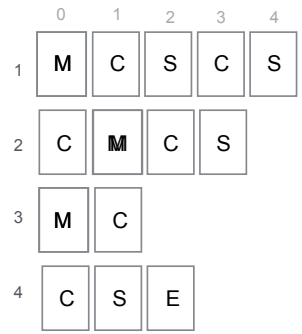
Improvements of Literature (3/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ? => reversible vectors** making use of **trailing techniques.**

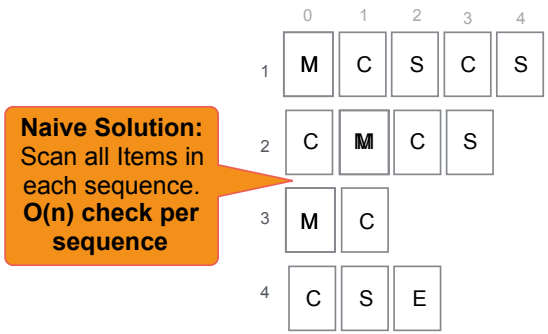
2. **Support Count Improvement. How to compute items support efficiently?** Visit **only the last position** of each symbol after **start position.**

3. **Sequence projection Improvement.** $P_1P_2P_3P_4P_5$ has just been extended to $EP_2P_3P_4P_5$. **Can we decide in O(1) if an embedding can not be extended?**.

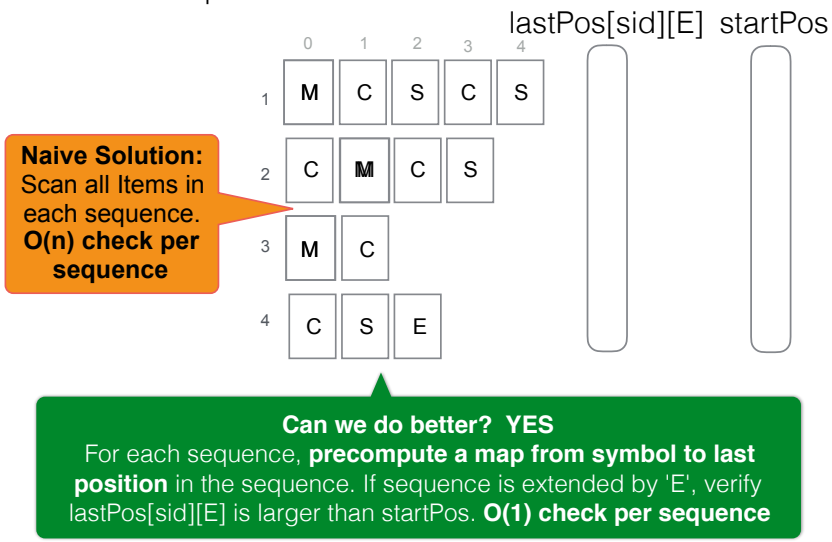
1. Which sequences contain at least one **E** ?



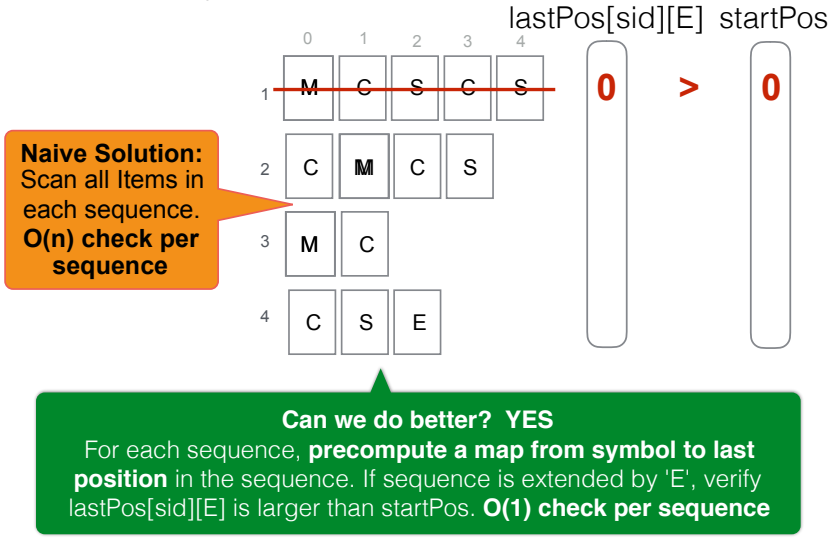
1. Which sequences contain at least one **E** ?



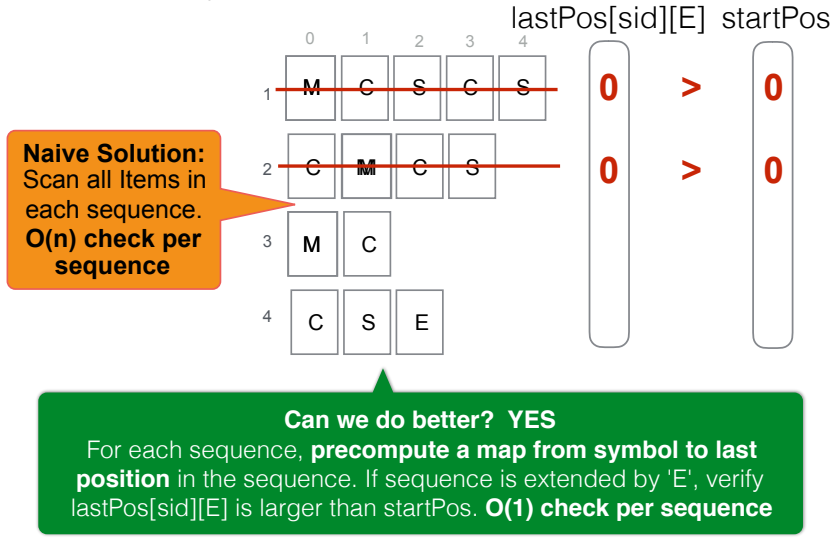
1. Which sequences contain at least one **E** ?



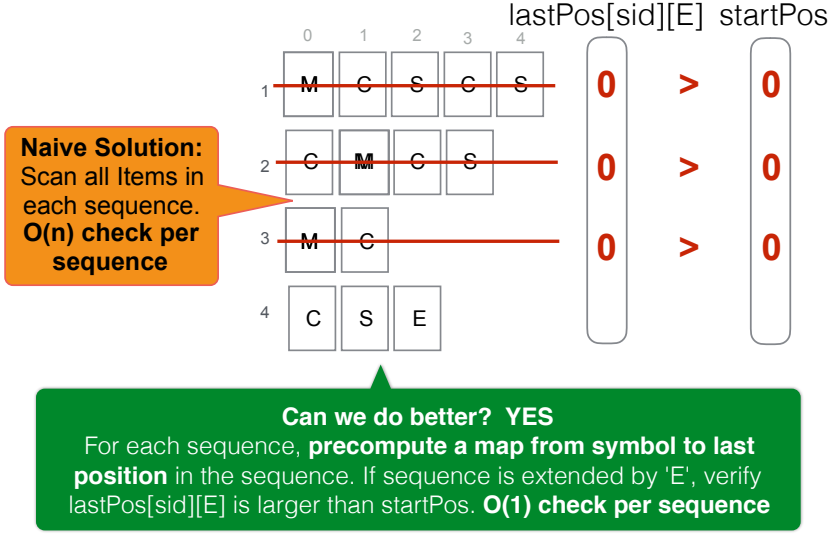
1. Which sequences contain at least one **E** ?



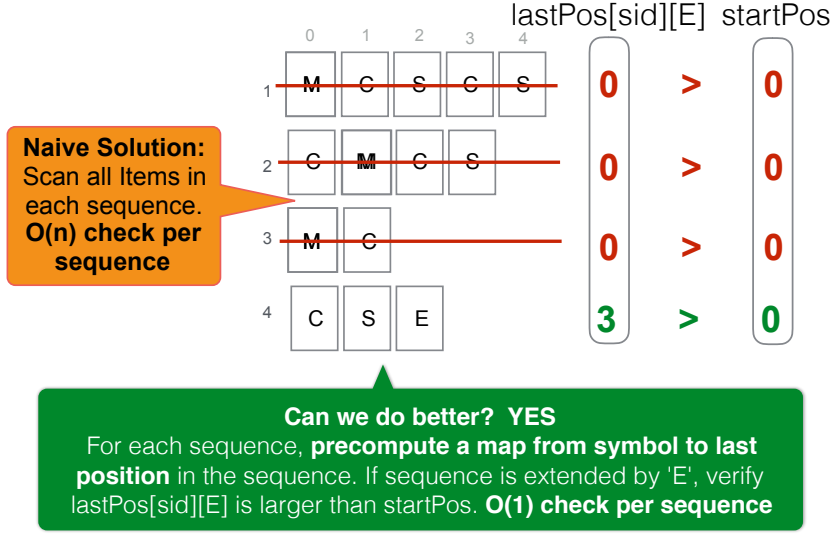
1. Which sequences contain at least one **E** ?



1. Which sequences contain at least one **E** ?



1. Which sequences contain at least one **E** ?



Improvements of Literature (4/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ? => reversible vectors** making use of **trailing techniques.**

2. **Support Count Improvement. How to compute items support efficiently?** Visit **only the last position** of each symbol after **start position..**

3. **Sequence visited Improvement.** $P_1P_2P_3P_4P_5$ has just been extended to $EP_2P_3P_4P_5$. **Can we decide in O(1) if an embedding can not be extended?**

4. **Pruning Improvement. Remove infrequent item from only D_{i+1} domains of P_{i+1}.**

Experiments

Results

Three Global constraints in OscaR implemented with Scala

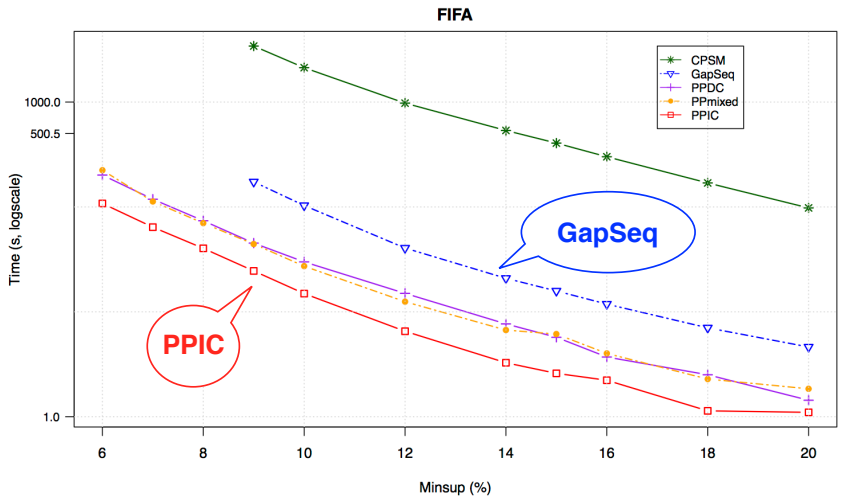
- **PPIC** : memory + support counting + sequence visited + prune improvements
- **PPDC** : same with PPIC but support counting is different.
- **PPmixed** : Mixed of PPIC/PPDC base on heuristic

http://sites.uclouvain.be/cp4dm/spm/

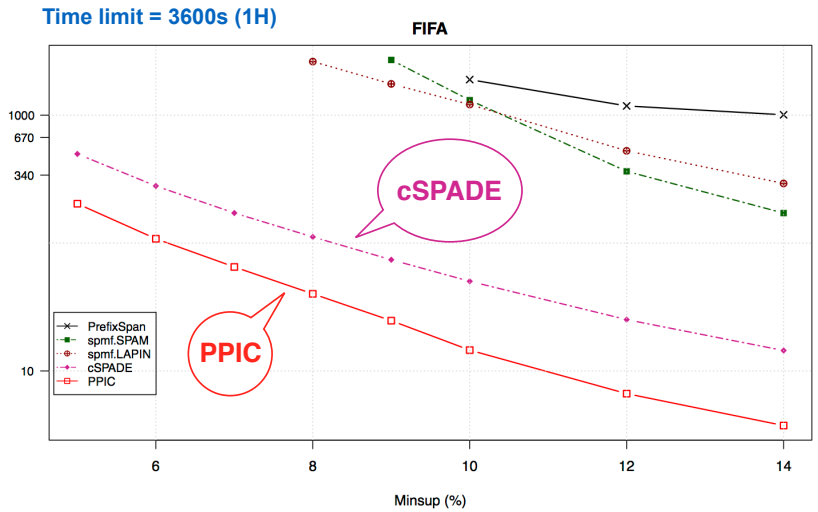
OSCAR www.oscarlib.org Scala

Compared with CP-Based methods

Time limit = 3600s (1H)

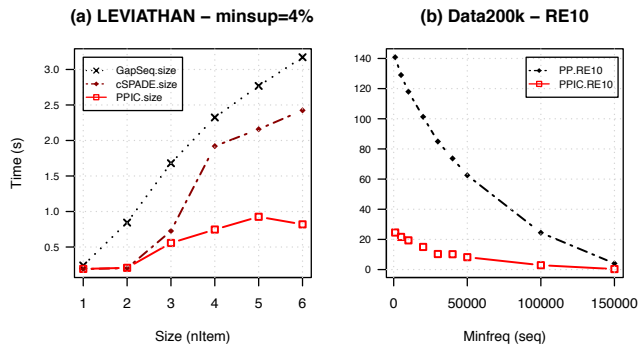


Compared with Specialized methods



Handling of different additional constraints

- Minimum and maximum pattern size constraint
- Item constraint
- Regular expression Contraint



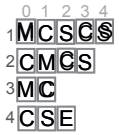
Conclusion

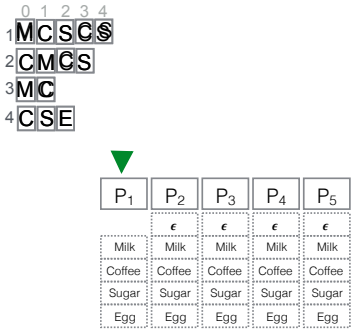
- Integrate ideas of both SPM and CP in a global constraint for CP solver (Generic and Flexible);
- First time, CP-based approach **outperforms both CP-based approaches and Specialized methods;**
- **Efficient memory using Trail-based backtracking aware datastructure** really speed up search in DFSearch;
- **Compatible with a number of other constraints, such as:** minimum and maximum length constraint, Regular expression constraint, etc
- **Future work:** use generality of framework for more constraints and other sequence mining settings (multi-objective, information theory based, interactive, ...)
- Code and apps are open **http://sites.uclouvain.be/cp4dm/spm/**

Improvements of Literature (1/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ?** => **reversible vectors** making use of **trailing techniques.**

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		





0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M :

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
E : 1

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
E : 1

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

Given
Threshold=3
(75%)

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

Given
Threshold=3
(75%)

M

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

Given
Threshold=3
(75%)

M

0 1 2 3 4
1 M C S C S
2 C M C S
3 M C
4 C S E

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

21

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1	C	S	C	S	
2		C	S		
3	C				

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0

21

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1	C	S	C	S	
2		C	S		
3	C				

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=0
Size=4

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

21

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1	C	S	C	S	
2		C	S		
3	C				

Supports
M : 3
C : 4
S : 3
~~E : 1~~

P_1	P_2	P_3	P_4	P_5
	ϵ	ϵ	ϵ	ϵ
Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg

start=4
Size=3

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

21

Given
Threshold=3
(75%)

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
		Sugar	Sugar	Sugar	Sugar
		Egg	Egg	Egg	Egg

Top of the sub-stack

start=0, size=4

TrailStack

start=4
Size=3

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

21

Given
Threshold=3
(75%)

Supports
M : 0
C : 3
S : 2
~~E : 1~~

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
		Sugar	Sugar	Sugar	Sugar
		Egg	Egg	Egg	Egg

Top of the sub-stack

start=0, size=4

TrailStack

start=4
Size=3

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

21

Given
Threshold=3
(75%)

Supports
~~M : 0~~
C : 3
~~S : 2~~
~~E : 1~~

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
		ϵ	ϵ	ϵ	ϵ
	Milk		Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
			Sugar	Sugar	Sugar
			Egg	Egg	Egg

Top of the sub-stack

start=0, size=4

TrailStack

start=4
Size=3

Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

21

Given
Threshold=3
(75%)

Supports
~~M : 0~~
C : 3
~~S : 2~~
~~E : 1~~

M

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

C

Supports
~~M : 0~~
C : 1
~~S : 2~~
~~E : 1~~

	0	1	2	3	4
1		C	S	C	S
2			C	S	
3		C			

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M : 3
C : 4
S : 3
~~E : 1~~

	P_1	P_2	P_3	P_4	P_5
			ϵ	ϵ	ϵ
	Milk		Milk	Milk	Milk
		Coffee	Coffee	Coffee	Coffee
			Sugar	Sugar	Sugar
			Egg	Egg	Egg

Top of the sub-stack

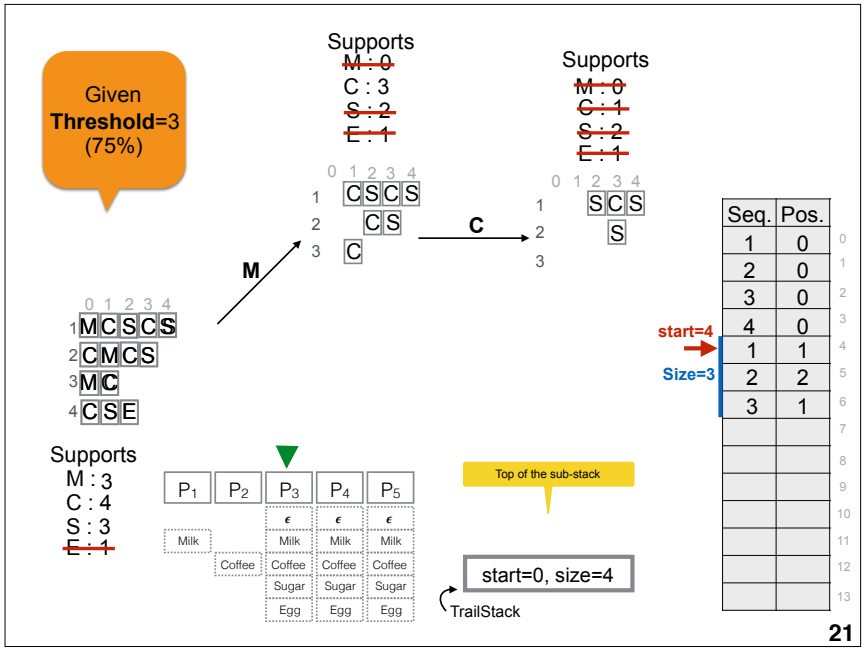
start=0, size=4

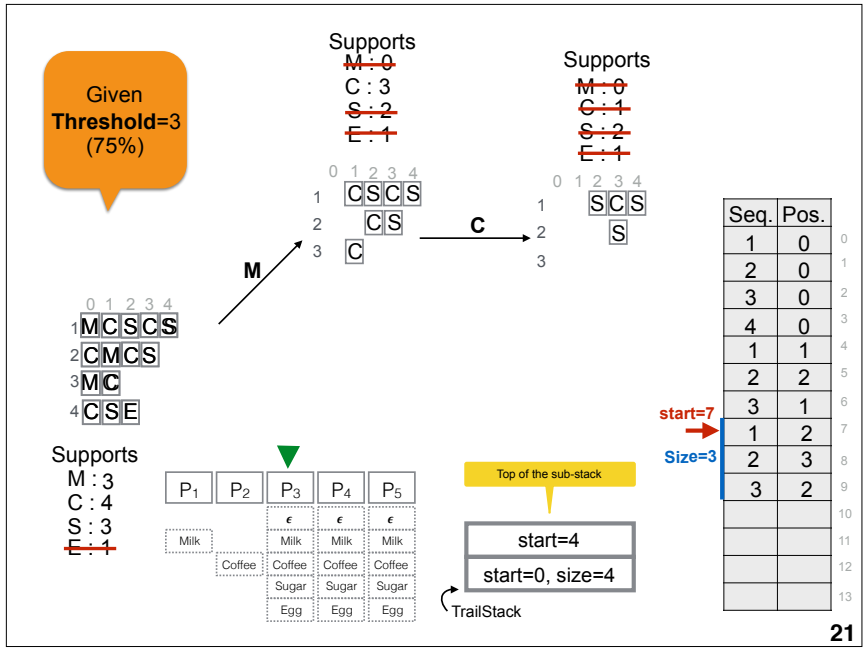
TrailStack

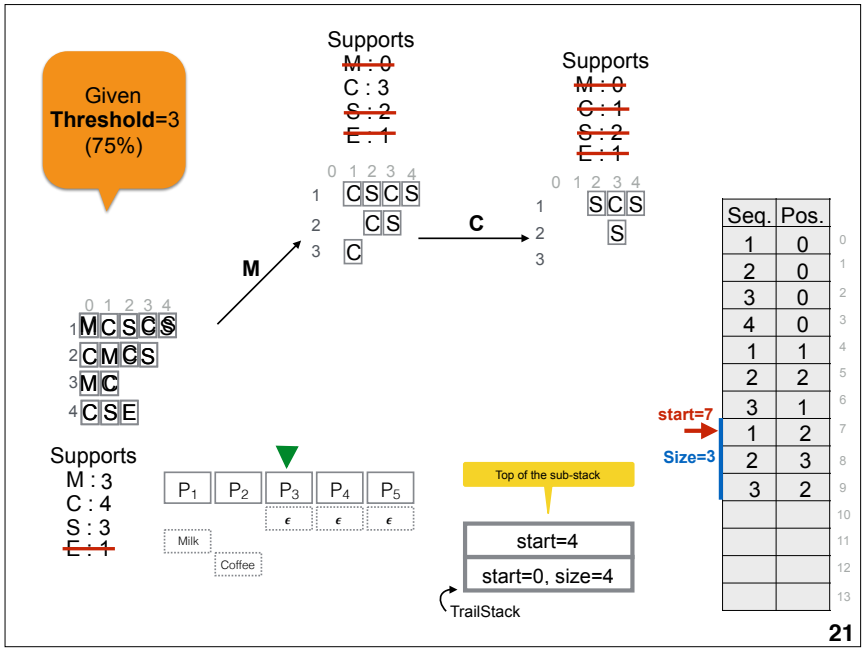
start=4
Size=3

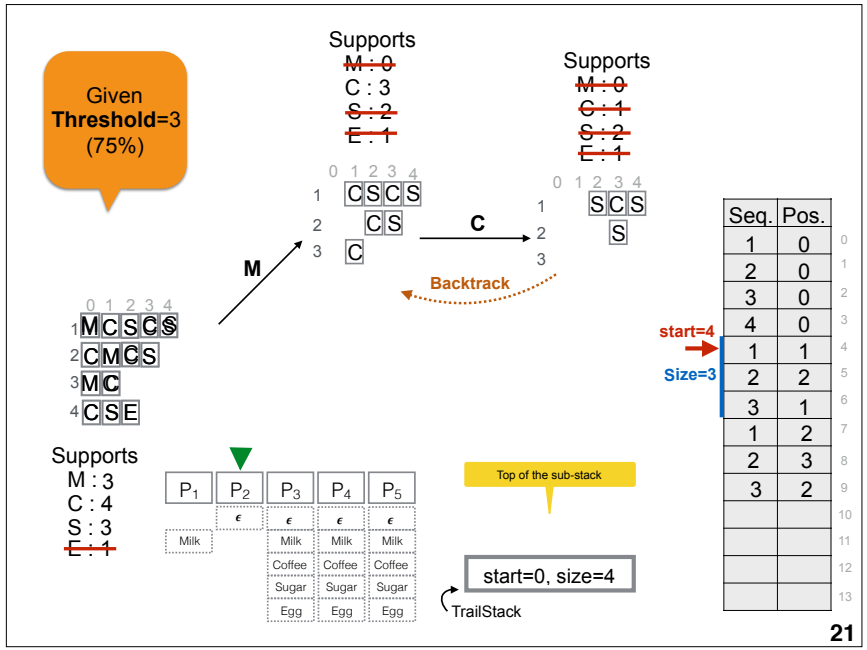
Seq.	Pos.
1	0
2	0
3	0
4	0
1	1
2	2
3	1

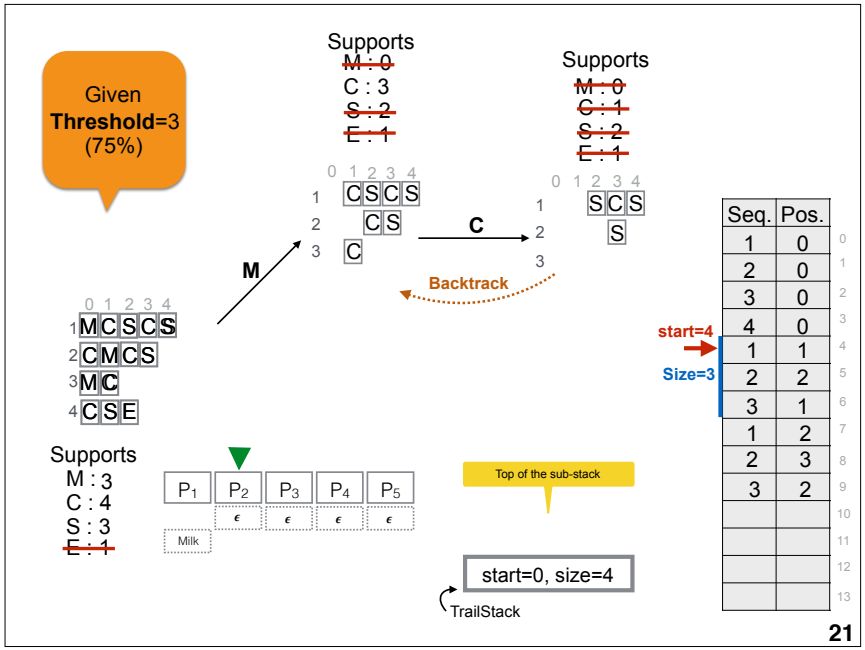
21

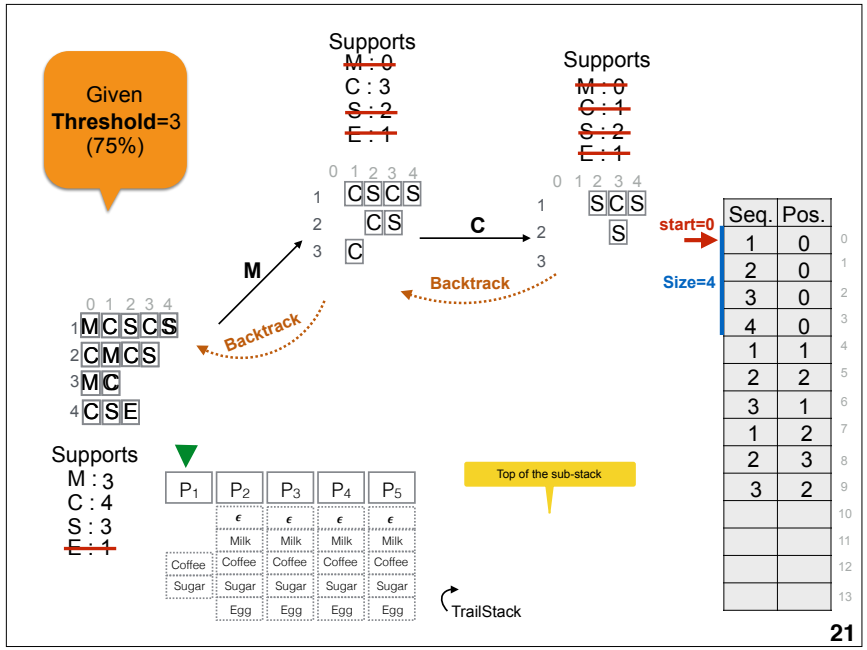

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
start=4
Size=3
Top of the sub-stack
start=0, size=4
TrailStack
21

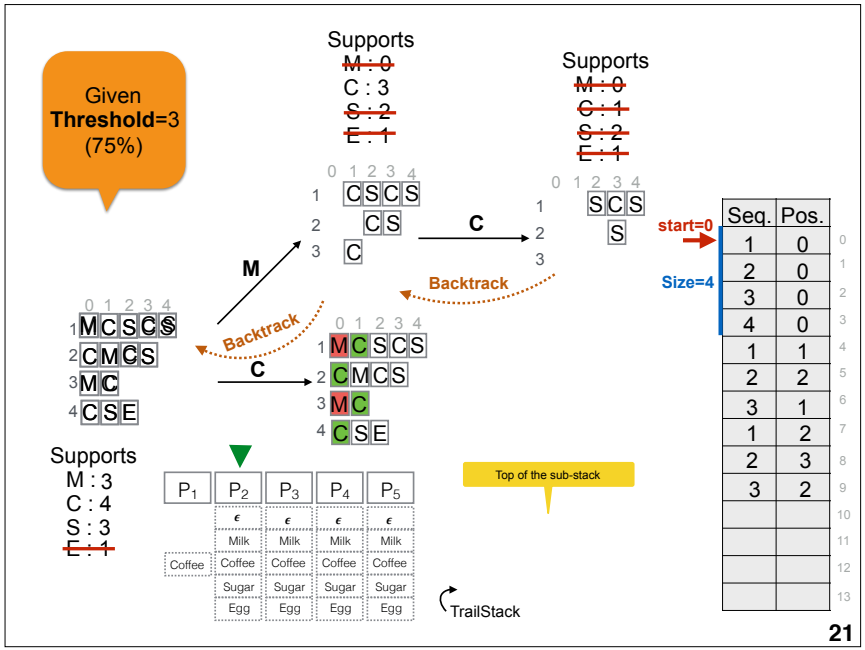

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
start=7
Size=3
Top of the sub-stack
start=4
start=0, size=4
TrailStack
21

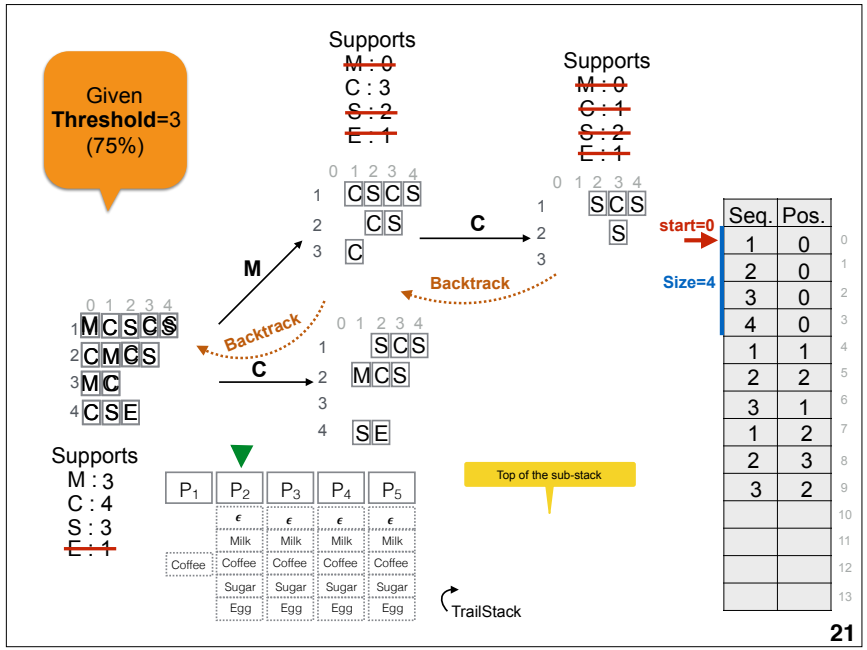

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
start=7
Size=3
Top of the sub-stack
start=4
start=0, size=4
TrailStack
21

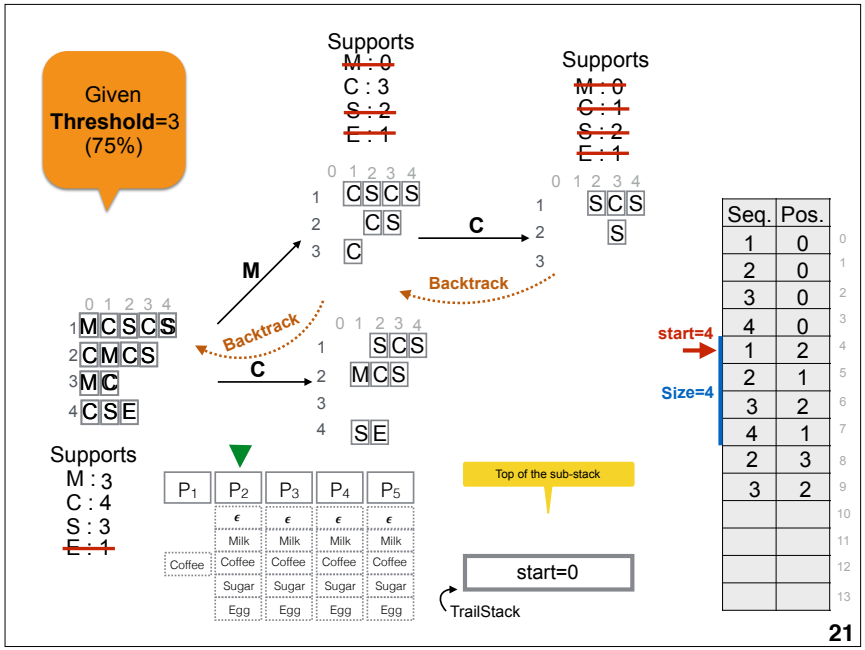

Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
start=4
Size=3
Top of the sub-stack
start=0, size=4
TrailStack
21


Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Top of the sub-stack
start=0, size=4
TrailStack
start=4
Size=3
Seq. Pos.
1 0
2 0
3 0
4 0
1 1
2 2
3 1
1 2
2 3
3 2
21


Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
Backtrack
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Coffee
Sugar
Egg
Top of the sub-stack
TrailStack
start=0
Size=4
Seq. Pos.
1 0
2 0
3 0
4 0
1 1
2 2
3 1
1 2
2 3
3 2
21


Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
Backtrack
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Coffee
Sugar
Egg
Top of the sub-stack
TrailStack
start=0
Size=4
Seq. Pos.
1 0
2 0
3 0
4 0
1 1
2 2
3 1
1 2
2 3
3 2
21


Given
Threshold=3
(75%)
Supports
M : 0
C : 3
S : 2
E : 1
Supports
M : 0
C : 1
S : 2
E : 1
M
C
Backtrack
Backtrack
C
M C S C S
C M C S
M C
C S E
Supports
M : 3
C : 4
S : 3
E : 1
P1 P2 P3 P4 P5
Milk
Coffee
Sugar
Egg
Top of the sub-stack
TrailStack
start=0
Size=4
Seq. Pos.
1 0
2 0
3 0
4 0
1 1
2 2
3 1
1 2
2 3
3 2
21



Improvements of Literature (2/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ? => reversible vectors** making use of **trailing techniques.**

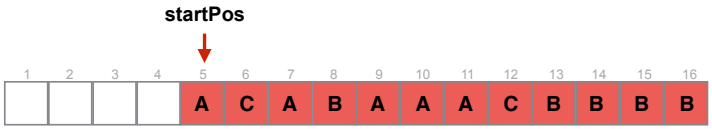
2. **Support Count Improvement. How to compute support efficiently?** Visit **only the last position** of each symbol after **start position.**

▶ How to compute items support in SDB?

★ Identify items which exist in each sequence of SDB and increase items support value.

▶ **Often many repeating symbols:** cache for each symbol its 'last position' and only iterate over those (**O(m) vs O(n)**)

✦ **startPos**=5

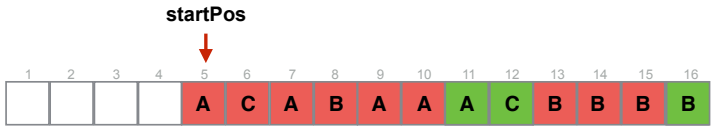


▶ How to compute items support in SDB?

★ Identify items which exist in each sequence of SDB and increase items support value.

▶ **Often many repeating symbols:** cache for each symbol its 'last position' and only iterate over those (**O(m) vs O(n)**)

✦ **startPos**=5



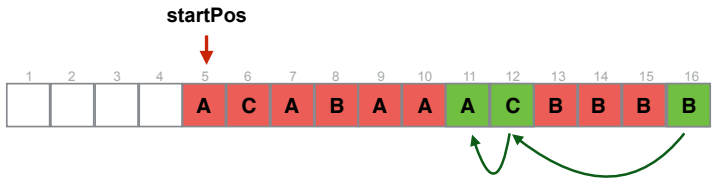
★ **Last Position List** = [(B,16),(C,12),(A,11),(D,2),(E,0)]

▶ How to compute items support in SDB?

★ Identify items which exist in each sequence of SDB and increase items support value.

▶ **Often many repeating symbols:** cache for each symbol its 'last position' and only iterate over those (**O(m) vs O(n)**)

✦ **startPos**=5



★ **Last Position List** = [(B,16),(C,12),(A,11),(D,2),(E,0)]

Improvements of Literature (3/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ?** => **reversible vectors** making use of **trailing techniques.**
2. **Support Count Improvement. How to compute items support efficiently?** Visit **only the last position** of each symbol after **start position.**
3. **Sequence projection Improvement.** $P_1P_2P_3P_4P_5$ has just been extended to $EP_2P_3P_4P_5$. **Can we decide in O(1) if an embedding can not be extended?**.

1. Which sequences contain at least one **E** ?

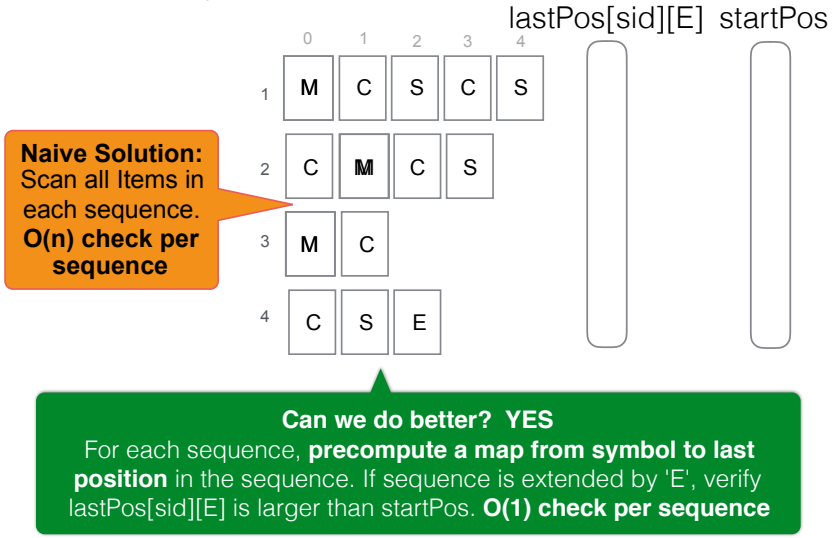
	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

1. Which sequences contain at least one **E** ?

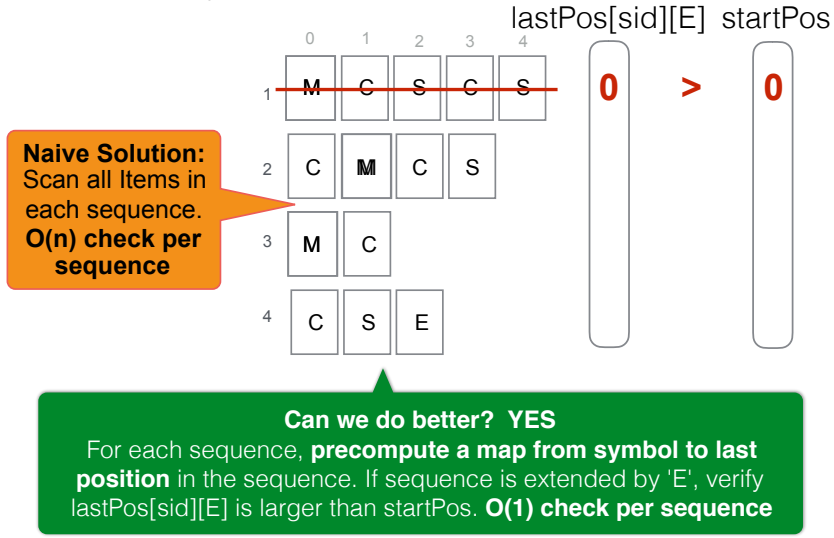
	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Naive Solution: Scan all Items in each sequence. **O(n) check per sequence**

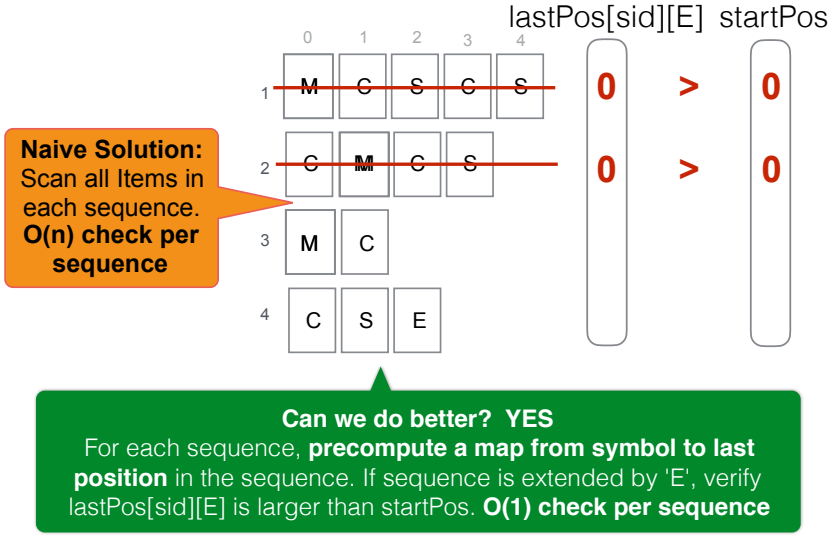
1. Which sequences contain at least one **E** ?



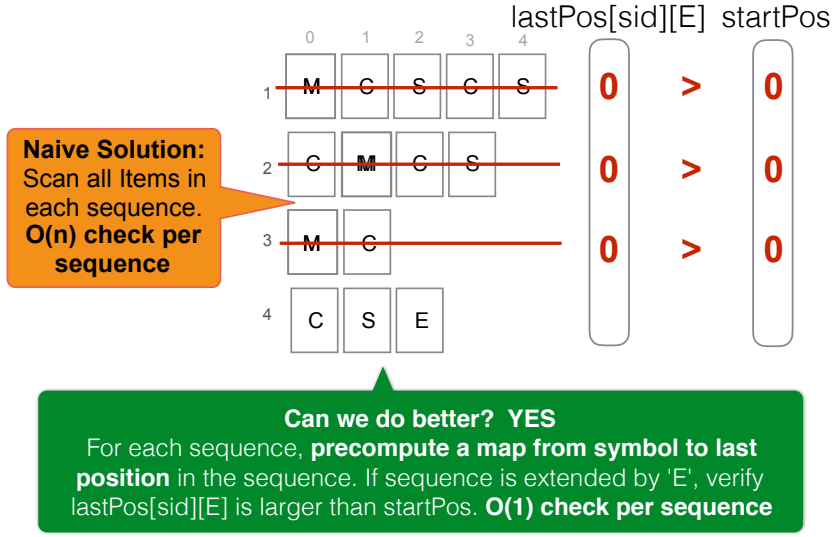
1. Which sequences contain at least one **E** ?



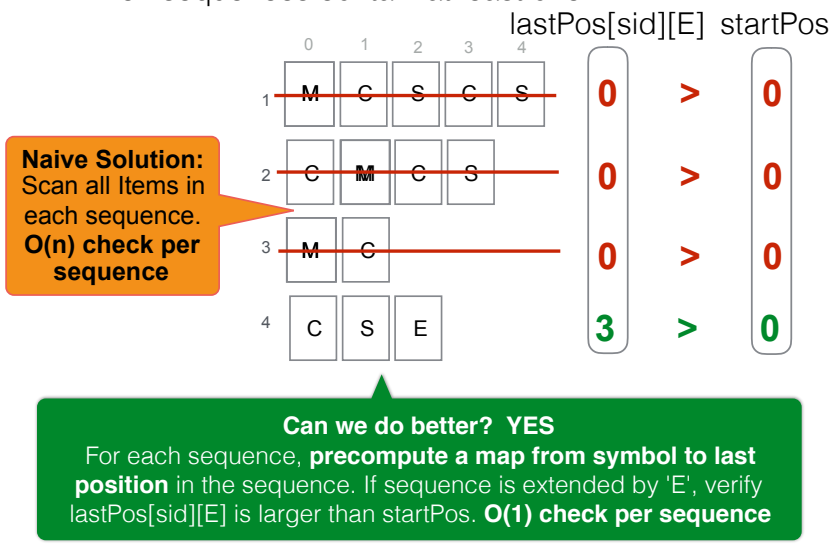
1. Which sequences contain at least one **E** ?



1. Which sequences contain at least one **E** ?



1. Which sequences contain at least one **E** ?



Improvements of Literature (4/4)

1. **Memory and DFS improvement. How to Store and restore databases in the DFSearch ? => reversible vectors** making use of **trailing techniques.**
2. **Support Count Improvement. How to compute items support efficiently?** Visit **only the last position** of each symbol after **start position..**
3. **Sequence visited Improvement.** $P_1P_2P_3P_4P_5$ has just been extended to $EP_2P_3P_4P_5$. **Can we decide in O(1) if an embedding can not be extended?**
4. **Pruning Improvement. Remove infrequent item from only D_{i+1} domains of P_{i+1}.**

Experiments

Results

Three Global constraints in OscaR implemented with Scala

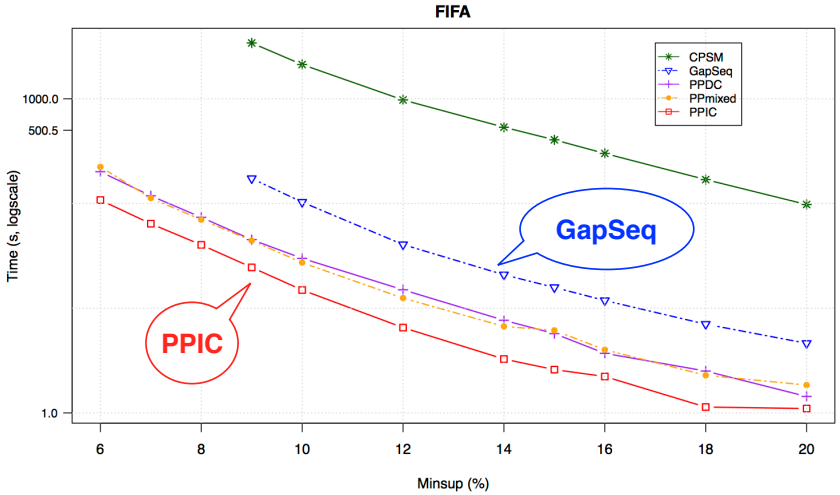
- **PPIC** : memory + support counting + sequence visited + prune improvements
- **PPDC** : same with PPIC but support counting is different.
- **PPmixed** : Mixed of PPIC/PPDC base on heuristic

http://sites.uclouvain.be/cp4dm/spm/

OSCAR www.oscarlib.org Scala

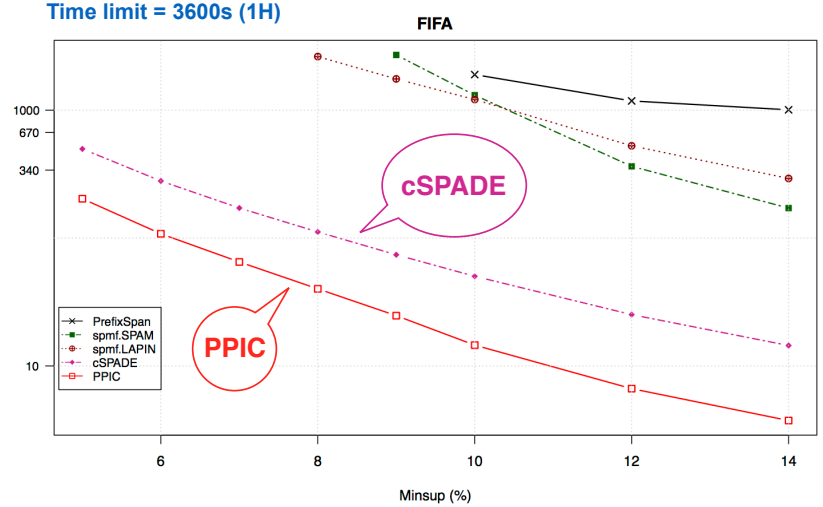
Compared with CP-Based methods

Time limit = 3600s (1H)



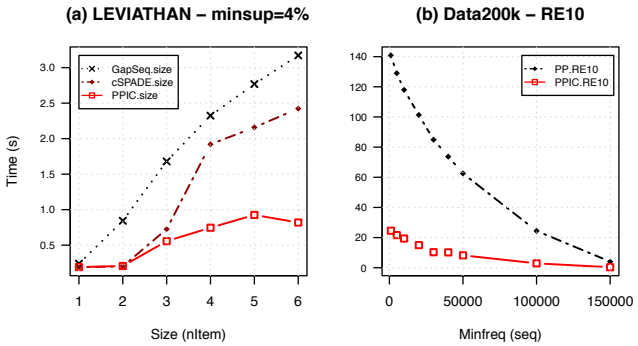
Compared with Specialized methods

Time limit = 3600s (1H)



Handling of different additional constraints

- Minimum and maximum pattern size constraint
- Item constraint
- Regular expression Contraint



Conclusion

- Integrate ideas of both SPM and CP in a global constraint for CP solver (Generic and Flexible);
- First time, CP-based approach **outperforms both CP-based approaches and Specialized methods;**
- **Efficient memory using Trail-based backtracking aware datastructure** really speed up search in DFSearch;
- **Compatible with a number of other constraints, such as:** minimum and maximum length constraint, Regular expression constraint, etc
- **Future work:** use generality of framework for more constraints and other sequence mining settings (multi-objective, information theory based, interactive, ...)
- Code and apps are open **http://sites.uclouvain.be/cp4dm/spm/**

thank you!